نظریه بازیها
Game Theory

ارائه کننده: امیرحسین نیکوفرد
مهندسی برق و کامپیوتر دانشگاه خواجه نصیر

Optimization

Material

- Convex Optimization, Stephen Boyd, Lieven Vandenberghe
  - Chapters 4.1 and 4.3

Optimization

Optimization problem:

$$\min_{x} \; f_o(x)$$

$$\textit{subject to}\,(s.t) \quad f_i(x) \le b_i, \quad i = \{1, \ldots, \mathrm{m}\}$$

The problem has several ingredients:

- The vector x collects the decision variables (optimization variables)
- $f_o(x) \quad \mathbb{R}^n \to \mathbb{R}$ objective function
- $f_i(x) \quad \mathbb{R}^n \to \mathbb{R}$ constraint functions

Optimal solution: x^* has smallest value of f_o among all vectors that satisfy the constraints

Solving optimization problems

general optimization problem

- very difficult to solve
- methods involve some compromise, e.g., very long computation time, or not always finding the solution

exceptions: certain problem classes can be solved efficiently and reliably

- least-squares problems
- linear programming problems
- convex optimization problems

Least-squares

$$\min_x \|Ax-b\|_2^2$$

solving least-squares problems

- Analytical solution: $x^* = (A^T A)^{-1} A^T b$
- reliable and efficient algorithms and software
- computation time proportional to $n^2 k$ ($A \in \mathbb{R}^{k\times n}$); less if structured
- a mature technology

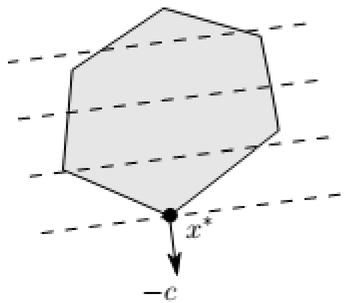
using least-squares

- least-squares problems are easy to recognize
- a few standard techniques increase flexibility (e.g., including weights, adding regularization terms)

Linear programming:

$$\min_{x} c^T x$$

$$s.t. \quad a_i^T x \le b_i, \quad i = \{1, \dots, m\}$$



solving linear programs

- no analytical formula for solution
- reliable and efficient algorithms and software
- computation time proportional to $n^2 m$ if $m > n$; less with structure
- a mature technology

using linear programming

- a few standard tricks used to convert problems into linear programs (e.g., problems involving ℓ_1-or ℓ_∞-norms, piecewise-linear functions)

Convex optimization problem

$$\min_{x} \; f_o(x)$$

$$\textit{subject to}\,(s.t) \quad f_i(x) \le b_i, \quad i = \{1, \ldots, \mathrm{m}\}$$

solving linear programs

- objective and constraint functions are convex:

$$f_i(\alpha x + \beta y) \le \alpha f_i(x) + \beta f_i(y)$$

If $\alpha + \beta = 1, \quad \alpha \ge 0, \beta \ge 0$

- includes least-squares problems and linear programs as special cases

Convex optimization problem

solving convex optimization problem

- no analytical formula for solution
- reliable and efficient algorithms
- computation time proportional to max$\{ n^3, n^2m, F\}$, where F is cost of evaluating f_i's and their first and second derivatives
- almost a technology

using convex optimization

- many tricks for transforming problems into convex form

Solving optimization problems

A simple optimization problem:

$$\min_{x \in R} |x_1 + 6| + |x_2 - 4|$$
$$s.t \quad 3 \le x_1 \le 5, \quad -2 \le x_2 \le 2$$

This problem is equivalent to a linear program (more on this later).

- Huge variety of software for solving LPs and QPs (and other standard types):
- Examples: MATLAB (linprog/quadprog), CPLEX, Gurobi, GLPK, XPRESS, qpOASES, OOQP, FORCES, SDPT3, Sedumi, MOSEK,,…
- There is no standard interface to solvers – they are almost all different.
- General purposes modeling tools allow easy switching between solvers: Examples: Yalmip ,CVX, GAMS, AMPL, TOMLAB,….

Solving optimization problems

V·T·E	Mathematical optimization software [hide]
Data formats	*LP* · *MPS* · *nl* · *OptML* · *OSiL* · *sol* · *xMPS*
Modeling tools	AIMMS · **AMPL** · APMonitor · CMPL · CVX · CVXOPT · CVXPY · ECLiPSe-CLP · GAMS · GNU MathProg · JuMP · LINDO · OPL · MPL · OptimJ · PICOS · PuLP · Pyomo · ROML · TOMLAB · Xpress-Mosel · YALMIP · ZIMPL
LP, MILP* solvers	ABACUS* · APOPT* · Artelys Knitro* · BCP* · BDMLP · BPMPD · BPOPT · CLP · CBC* · CPLEX* · CSDP · DSDP · FortMP* · GCG* · GIPALS32 · GLPK/GLPSOL* · Gurobi* · HOPDM · LINDO* · lp_solve* · LOQO · MINOS · MINTO* · MOSEK* · OOPS · OOQP · PCx · QSopt · SAS/OR* · SCIP* · SoPlex · SOPT-IP* · Sulum Optimization Tools* · SYMPHONY* · XA* · Xpress-Optimizer*
QP, MIQP* solvers	APOPT* · Artelys Knitro* · BPMPD · BPOPT · BQPD · CBC* · CLP · CPLEX* · FortMP* · GloMIQO* · Gurobi* · IPOPT · LINDO* · LSSOL · LOQO · MINOS · MOSEK* · OOPS · OOQP · QPOPT · QPSOL · SCIP* · XA Quadratic Solver · Xpress-Optimizer*
QCP, MIQCP* solvers	APOPT* · Artelys Knitro* · BPMPD · BPOPT · CPLEX* · GloMIQO* · Gurobi* · IPOPT · LINDO* · LOQO · MINOS · MOSEK* · SCIP* · Xpress-Optimizer* · Xpress-SLP*
SOCP, MISOCP* solvers	CPLEX* · DSDP · Gurobi* · LINDO* · LOQO · MOSEK* · SCIP* · SDPT3 · SeDuMi · Xpress-Optimizer*
SDP, MISDP* solvers	CSDP · DSDP · MOSEK · PENBMI · PENSDP · SCIP-SDP* · SDPA · SDPT3 · SeDuMi
NLP, MINLP* solvers	ALGENCAN · AlphaECP* · ANTIGONE* · AOA* · APOPT* · Artelys Knitro* · BARON* · Bonmin* · BPOPT · CONOPT · Couenne* · DICOPT* · FilMINT* · FilterSQP · Galahad library · ipfilter · IPOPT · LANCELOT · LINDO* · LOQO · LRAMBO · MIDACO* · MILANO* · MINLP BB* · MINOS · Minotaur* · MISQP* · NLPQLP · NPSOL · OQNLP* · PATHNLP · PENNON · SBB* · SCIP* · SNOPT* · SQPlab · WORHP · Xpress-SLP*
GO solvers	BARON · Couenne* · LINDO · SCIP
CP solvers	Artelys Kalis · Choco · Comet · CPLEX CP Optimizer · Gecode · Google CP Solver · JaCoP · OscaR
Metaheuristic solvers	OptaPlanner · LocalSolver
	List of optimization software · Comparison of optimization software

Solving optimization problems

A simple optimization problem:

$$\min_{x \in R} |x_1 + 6| + |x_2 - 4|$$

$$s.t \quad 3 \le x_1 \le 5, \quad -2 \le x_2 \le 2$$

The **YALMIP toolbox** for Matlab (from ETH / Linkoping):

```
%make variables
sdpvar x1 x2;
%define cost function
f = abs(x1 + 6) + abs(x2 - 4);
%define constraints
X = set(3 <= x1 <= 5) + ...
set( -2 <= x2 <= 2);
%solve
solvesdp(X,f);
```

Solving optimization problems

A simple optimization problem:

$$\min_{x \in R} |x_1 + 6| + |x_2 - 4|$$

$$s.t \quad 3 \le x_1 \le 5, \quad -2 \le x_2 \le 2$$

The **CVX toolbox** for Matlab (from Stanford):

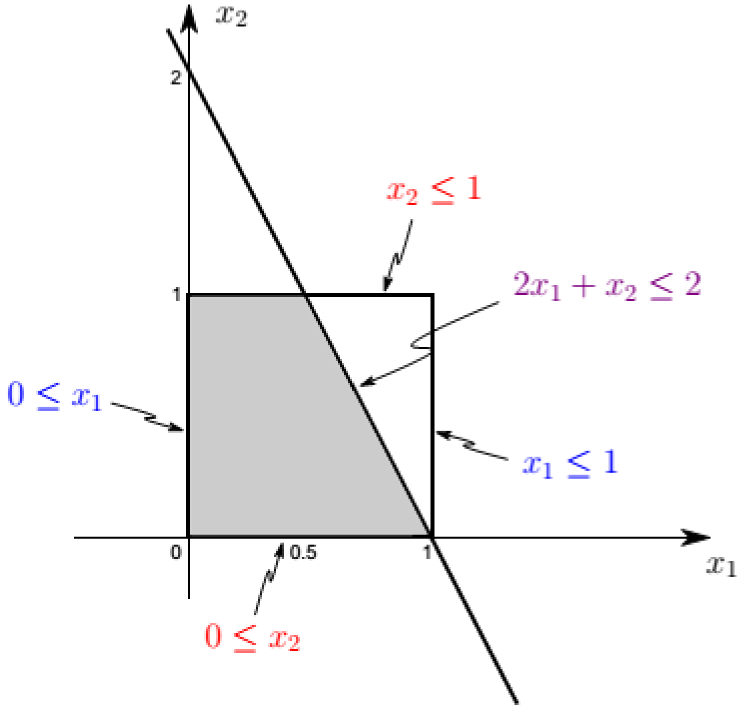
```
cvx_begin
variables x1 x2 % define variables
%define cost function and constraints
minimize(abs(x1 + 6) + abs(x2 - 4))
subject to
3 <= x1 <= 5
-2 <= x2 <= 2
cvx _end %solves automatically
```

Linear Program (LP)

- The optimal solution of LP lies one of the corner points or facets of the feasible region

$$\max_{x_1,x_2} x_1 + x_2$$

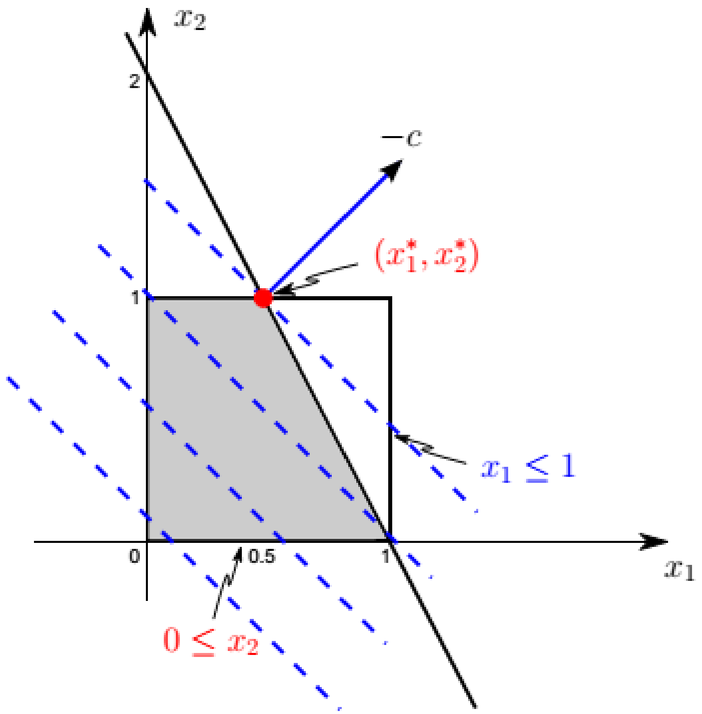
$$\begin{aligned} s.t. \quad & 0 \le x_1 \le 1 \\ & 0 \le x_2 \le 1 \\ & 2x_1 + x_2 \le 2 \end{aligned}$$



Linear Program (LP)

❑ The optimal solution of LP lies one of the corner points or facets of the feasible region

$$\max_{x_1,x_2} x_1 + x_2$$
$$s.t. \quad 0 \le x_1 \le 1$$
$$0 \le x_2 \le 1$$
$$2x_1 + x_2 \le 2$$



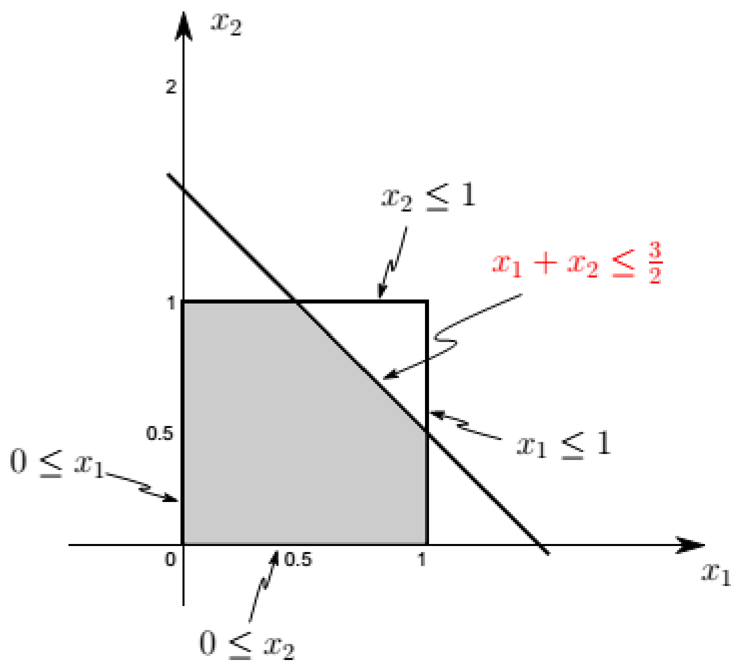
Vector c= (−1−1)(change maximization to minimization)

Optimal solution: $(x_1^*, x_2^*) = (0.5,1)$

Linear Program (LP)

- The optimal solution of LP lies one of the corner points or facets of the feasible region

$$
\begin{aligned}
\max_{x_1, x_2}\ & x_1 + x_2 \\
s.t.\quad & 0 \le x_1 \le 1 \\
& 0 \le x_2 \le 1 \\
& x_1 + x_2 \le 3/2
\end{aligned}
$$



Linear Program (LP)

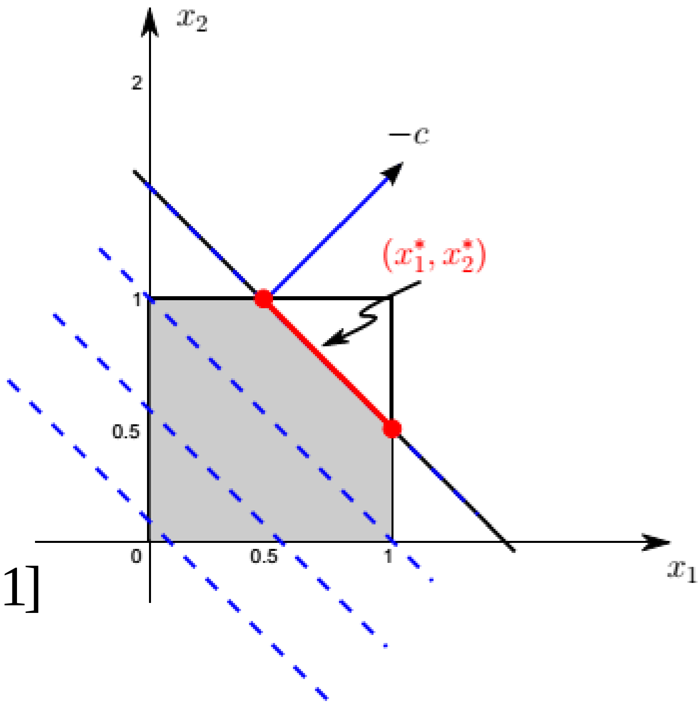
❑ The optimal solution of LP lies one of the corner points or facets of the feasible region

$$\max_{x_1, x_2} x_1 + x_2$$

$$s.t. \quad 0 \le x_1 \le 1$$

$$0 \le x_2 \le 1$$

$$x_1 + x_2 \le 3/2$$



Optimal solution not unique:

$$(x_1^*, x_2^*) = \lambda(0.5,1) + (1-\lambda)(1,0.5) \quad \lambda \in [0,1]$$

Optimizing over simplexes

A simplex is defined as follows:

$$X := \{x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1,\ x_i \geq 0,\ i = 1, \ldots, n\}$$

- Consider the optimization problem:

$$\min_x \sum_{j=1}^{n} x_j \beta_j \qquad \Leftrightarrow \min_{j \in \{1,\ldots,n\}} \beta_j$$
$$s.t.\ \ x \in X$$

- Optimal x^* lies in the corner point of simplex X. Similarly,

$$\max_x \sum_{j=1}^{n} x_j \beta_j \qquad \Leftrightarrow \max_{j \in \{1,\ldots,n\}} \beta_j$$
$$s.t.\ \ x \in X$$

Epigraph problem form

The epigraph form of the standard optimization problem is the problem

$$\min_{x,t} t$$

$$s.t. \quad f_o(x) \le \mathrm{t} \quad t \in \mathbb{R}, x \in \mathbb{R}^n$$

$$f_i(x) \le b_i, \quad i = \{1, \ldots, \mathrm{m}\}$$

- We can easily see that it is equivalent to the original problem:(x,t) is optimal for the epigraph form if and only if x is optimal for the standard optimization problem and $t = f_o(x)$. Note that the objective function of the epigraph form problem is a linear function of the variables x, t.

Example Linear Programs

Piecewise affine minimization

$$\min_x \left[\max_{i=1,\dots,m} \{c_i x + d_i\} \right]$$

$$s.t. \quad Ax \le b$$

is equivalent to an LP:

$$\min_{x,t} \quad t$$

$$s.t. \quad c_i x + d_i \le t \quad i = \{1,\dots,m\}$$

$$Ax \le b$$

- trick was to add variables and write the problem in epigraph form.

Recap :Mixed Strategies

In mixed strategies:

- the players select their actions randomly according to a previously selected probability distribution

A mixed policy for P_1 is a set of numbers

$$Y=\{(y_1,\ldots,y_m):\sum_{i=1}^{m} y_i = 1,\ y_i \geq 0,\ i=1,\ldots,m\ \}$$

- A mixed policy for P_2 is a set of numbers

$$Z=\{(z_1,\ldots,z_n):\sum_{j=1}^{n} z_j = 1,\ z_j \geq 0,\ j=1,\ldots,n\ \}$$

Column player's actions

Row player's actions

	z_1	z_2	…	z_n
y_1	a_{11}	a_{12}	…	a_{1n}
y_2	a_{21}	a_{22}	…	a_{2n}
	…			
y_m	a_{m1}	a_{m2}	…	a_{mn}

Recap: Mixed Strategies

Definition: (Mixed saddle-point equilibrium):

A pair of policies defined through the probabilities $(y^*, z^*) \in Y \times Z$ is called a mixed saddle-point equilibrium if

$$y^{*T} A z^* \geq y^T A z^* \qquad \forall y \in Y \qquad \text{(the maximizer)}$$

$$y^{*T} A z^* \leq y^{*T} A z \qquad \forall z \in Z \qquad \text{(the minimizer)}$$

and $y^{*T} A z^*$ is the is called the saddle point value.

This is also called a Nash Equilibrium: no player can do better by **unilaterally** changing his strategy (includes both pure and mixed strategies)

Recap: Mixed Strategies

Theorem (Mixed saddle-point vs. security levels)

A matrix game defined by A has a **mixed saddle-point equilibrium** if and only if

$$\underline{V}_m(A) = \max_{y \in Y} \min_{z \in Z} y^T A z = \min_{z \in Z} \max_{y \in Y} y^T A z = \overline{V}_m(A)$$

In particular,

- (y^*, z^*) is a mixed saddle-point equilibrium
- $\underline{V}_m(A) = \overline{V}_m(A)$ is the saddle point value

This condition holds for all matrices A

For any two **player zero-sum game** there exists a saddle point equilibrium (Nash equilibrium)

- You can check this condition by just checking $\underline{V}_m(A) = \overline{V}_m(A)$

Computing Mixed Strategies

To compute the mixed security policy for P$_1$ (the maximizer) we need to solve

$$y^* \in \arg\max_{y \in Y} \min_{z \in Z} y^T A z$$

To compute the mixed security policy for P$_2$ (the minimizer) we need to solve

$$z^* \in \arg\min_{z \in Z} \max_{y \in Y} y^T A z$$

Optimizing over simplexes

A simplex is defined as follows:

$$X := \{x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1,\ x_i \geq 0,\ i = 1, \ldots, n\ \}$$

- Consider the optimization problem:

$$\min_x \sum_{j=1}^{n} x_j \beta_j \quad \Leftrightarrow \quad \min_{j \in \{1,\ldots,n\}} \beta_j$$
$$s.t.\ \ x \in X$$

- Optimal x^* lies in the corner point of simplex X. Similarly,

$$\max_x \sum_{j=1}^{n} x_j \beta_j \quad \Leftrightarrow \quad \max_{j \in \{1,\ldots,n\}} \beta_j$$
$$s.t.\ \ x \in X$$

Computing Mixed Strategies

Computing the security strategy for P_1 (the maximizer)

$$\underline{V}_m(A) := \max_{y \in Y} \min_{z \in Z} y^T A z = \max_{y \in Y} \min_{z \in Z} \sum_{i=1}^{m} \sum_{j=1}^{n} z_j y_i a_{ij}$$

Remembering that Z is a simplex, if we look at the inner minimization

$$\min_{z \in Z} \sum_{i=1}^{m} \sum_{j=1}^{n} z_j y_i a_{ij} \xrightarrow{Z \text{ is a simplex}} \min_{j \in \{1,\ldots,n\}} \sum_{i=1}^{m} y_i a_{ij}$$

$$= \min\{ \sum_{i=1}^{m} y_i a_{i1}, \ldots \sum_{i=1}^{m} y_i a_{im} \}$$

Computing Mixed Strategies

To compute the mixed security policy for P_1 we need to solve

$$\underline{V}_m(A) := \max_{y \in Y} \min_{z \in Z} y^T A z = \max_{y \in Y} \min\{\sum_{i=1}^{m} y_i a_{i1}, \ldots \sum_{i=1}^{m} y_i a_{im}\}$$

- The resulting problem requires maximizing a **convex piecewise linear function** which is equivalent to the following linear program

$$\underline{V}_m(A) := \max_{y,t} \ t$$

$$s.t. \sum_{i=1}^{m} y_i a_{ij} \geq t \qquad j = 1, \ldots, n$$

$$y \in Y, t \in \mathbb{R}$$

Piecewise affine maximization

Piecewise affine maximization

$$\max_{x} \left[\min_{i=1,\ldots,m} \{c_i x + d_i\} \right]$$

$$s.t. \quad Ax \le b$$

is equivalent to an LP:

$$\max_{x,t} \quad t$$

$$s.t. \quad c_i x + d_i \ge t \quad i = \{1,\ldots,m\}$$

$$Ax \le b$$

- ❑ trick was to add variables and write the problem in epigraph form.

Computing Mixed Strategies

- In compact form,

$$\begin{aligned} \underline{V}_m(A) := \max_{y,t} \ & t \\ s.t. \ & A^T y \geq t \\ & 1^T y = 1 \\ & y \geq 0 \\ & y \in \mathbb{R}^m, t \in \mathbb{R} \end{aligned}$$

- Where $1 = (1, \ldots, 1)^T \in \mathbb{R}^m$ is the vector ones

Computing Mixed Strategies

- Following similar arguments, we can show that P_2 can compute his security strategy solving the following linear optimization problem

$$\bar{V}_m(A) := \min_{z,t} \ t$$

$$\begin{aligned} s.t.\ & Az \leq t \\ & 1^T z = 1 \\ & z \geq 0 \\ & z \in \mathbb{R}^n, t \in \mathbb{R} \end{aligned}$$

- Where $1 = (1, \ldots, 1)^T \in \mathbb{R}^n$ is the vector ones

Computing Mixed Strategies

- Yalmip code for solving the LP for P_2

$$
\begin{aligned}
\bar{V}_m(A) := \min_{z,t} \ & t \\
s.t.\ & Az \le t \\
& 1^T z = 1 \\
& z \ge 0 \\
& z \in \mathbb{R}^n, t \in \mathbb{R}
\end{aligned}
$$

Computing Mixed Strategies

- Yalmip code for solving the LP for P_2

```
m = 5; n = 10; % Define the matrix defining the zero-sum game
A = rand(m,n);
z = sdpvar(n,1); % Define optimization variables
t = sdpvar(1,1);
obj = t; % Define objective function
% Define constraints
constraints = [A*z <= ones(m,1)*t, sum(z) == 1, z>= 0];
optimize(constraints,obj); % Solve optimization problem
SecurityLevel = double(t) % Get solution
SecurityPolicy = double(z)
```